P516762 PDF [Page: 3 of 70]

Copy No. 11

DEFENCE R&D DÉFENSE

FINAL REPORT ON THE PORT OF SIGNAL PROCESSING SOFTWARE TO LINUX

D. R. Chang
Evans Computer Applications Ltd.
6424 Norwood St.
Halifax, N.S. B3H 2L3

Defence R&D Canada
DEFENCE RESEARCH ESTABLISHMENT ATLANTIC

Contractor Report
DREA CR 2001-108
August 2001

National Defence Défense nationale

Canada

P516762.PDF [Page: 4 of 70]

104-06995/

Copy No: _____

Final Report on the Port of Signal Processing Software to Linux

David R. Chang

Evans Computer Applications Ltd.
6424 Norwood St.
Halifax NS B3H 2L3

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Defence Research Establishment Atlantic

Contractor Report

DREA CR 2001-108

2001-08-09

Author

David R. Chang

Approved by

A/ H/ I. A. Fraser

Maritime Information Systems

Approved for release by

Kirk Foster

Chair / Document Review Panel

© Her Majesty the Queen as represented by the Minister of National Defence, 2001

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2001

P516762.PDF [Page: 6 of 70]

Abstract

This report describes how the Red Hat Linux 6.0 operating system was installed on a Windows NT system at DREA. The subsequent port of some signal processing software to Linux, including SAPLOTX, PRODAT, PLTPWR, PROPWR, DISPX, HETFIR, and COPYDAT, is also outlined. Additionally, the report describes the difficulties encountered while performing these tasks, and the issues that still remain to be resolved.

As an auxiliary task, Red Hat Linux 7.0 was installed on a Windows 2000 system. This activity is also fully documented.

Résumé

Le présent rapport décrit la façon dont le système d'exploitation Red Hat Linux 6.0 a été installé sur un système Windows NT au CRDA. Y est décrite également l'adaptation à Linux de certains logiciels de traitement du signal, tels que SAPLOTX, PRODAT, PLTPWR, PROPWR, DISPX, HETFIR et COPYDAT. On y trouvera d'autre part les difficultés rencontrées pendant l'exécution de ces tâches et les problèmes qui restent encore à résoudre.

L'installation du Red Hat Linux 7.0 a été effectuée sur un système Windows 2000 comme tâche secondaire. L'activité est également documentée en entier.

P516762.PDF [Page: 7 of 70]

This page intentionally left blank.

Executive summary

Introduction

Because UNIX-based computer operating systems are considerably less demanding on system resources than Windows systems, UNIX has long been the platform of choice for scientific data analysis. In particular, programs running on the SGI IRIX operating system have been used extensively at DREA to carry out acoustical research. Unfortunately, due to binary incompatibility between different variants of UNIX, programs that are available to use on one system may not operate on another. When the decision was made at DREA to standardize to dual-boot Windows NT / Linux workstations, several programs (including the greyscaling and power-spectra analysis tools PRODAT and PLTPWR, the file transfer utility COPYDAT, and the time series data plotter DISPX) were unable to operate in the new Linux environment. This contractor's report documents the steps taken to bring this essential scientific software to Linux.

Results

This document contains an account of the installation of Red Hat Linux 6.0 and associated software on a DELL OptiPlex GX1 workstation. It also covers alterations made to several function libraries and programs to allow successful operation within the Linux environment. The installation of Red Hat Linux 7.0 on a Windows 2000 desktop computer, an auxiliary task that was also performed and recorded, adds to a growing database of Linux-related knowledge and experience.

Significance

The Linux operating system is perhaps the most successful product to date of the "open source" software movement, with an enormous, rapidly-growing support base of programmers and engineers from all over the world. Because it is an economical and readily available UNIX variant, the Linux platform lends itself well to scientific applications. The availability of sonar signal processing software designed for this operating system opens new and inexpensive options for computing in the future.

Chang, David R. 2001. Final Report on the Port of Signal Processing Software to Linux. DREA CR 2001-108. Evans Computer Applications Ltd.

P516762.PDF [Page: 9 of 70]

Sommaire

Introduction

Étant donné que les systèmes d'exploitation d'ordinateur à base UNIX exigent considérablement moins de ressources de système que les systèmes Windows, UNIX a depuis longtemps été la plate-forme de choix pour l'analyse de données scientifiques. Au CRDA, on s'est en particulier beaucoup servi de programmes qui s'exécutent sur le système d'exploitation SGI IRIX pour effectuer des recherches acoustiques. Malheureusement, à cause d'une incompatibilité entre les variantes d'UNIX, les programmes qu'on peut utiliser sur un système peuvent ne pas fonctionner sur un autre. Lorsqu'on a pris la décision de normaliser le lancement des postes de travail Windows NT / Linux, plusieurs programmes (notamment le PRODAT et le PLTPWR, outils servant à l'analyse spectrale de puissance et à la réduction des gris, le COPYDAT, utilitaire de transfert de fichiers et le DISPX, traceur de séries chronologiques) ne pouvaient pas fonctionner dans le nouvel environnement Linux. Ce rapport d'entrepreneur étaie les étapes suivies pour adapter à Linux ce logiciel scientifique indispensable.

Résultats

Ce document renferme un compte rendu de l'installation du Red Hat Linux 6.0 et des logiciels connexes sur un poste de travail DELL OptiPlex GXI. Il fait état aussi des modifications qui ont été apportées à plusieurs bibliothèques et programmes de fonction pour assurer un bon fonctionnement dans un environnement Linux. L'installation du Red Hat Linux 7.0 sur un ordinateur de bureau doté de Windows 2000, tâche secondaire effectuée et enregistrée, renforce une base grandissante de connaissances et d'expérience liées à Linux.

Portée

Le système d'exploitation Linux est peut-être à ce jour le produit le plus réussi parmi le flux de logiciels « de source ouverte », jouissant d'une base de soutien formée de programmeurs et d'ingénieurs provenant du monde entier et dont le nombre grandit rapidement. Variante économique et pratique d'UNIX, la plate-forme Linux se prête bien aux applications scientifiques. Le fait qu'on dispose d'un logiciel de traitement du signal sonar conçu pour ce système d'exploitation permet de considérer de nouvelles options informatiques peu coûteuses pour l'avenir.

Chang, David R. 2001. Rapport final sur l'adaptation du logiciel de traitement du signal sonar à Linux. DREA CR 2001-108. Evans Computer Applications Ltd.

Table of contents

Abstract i

Résumé i

Executive summary iii

Sommaire iv

Table of contents v

List of Tables viii

I. INTRODUCTION 1

II. INSTALLING RED HAT LINUX 6.0 2

1. PREPARING FOR INSTALLATION 2

2. PARTITIONING THE DISKS 2

2.1 Performing the Partitioning 3

3. INSTALLING LINUX 3

3.1 Creating Partitions for Linux (pp. 48 - 57 of guide) 3

3.2 Filesystem Configuration (p. 57 of guide) 4

3.3 Selecting Components (p. 61 of guide) 4

3.4 Package Installation (p. 64 of guide) 5

3.5 Configuring the X Window System (pp. 108-109 of guide) 5

4. LINUX CONFIGURATION 5

4.1 Allowing an Ordinary User to Mount the Floppy Disk 5

4.2 Allowing an Ordinary User to Access the Tape 5

4.3 Allowing Remote FTP Access to the Computer 6

4.4 Configuring a Printer 6

4.5 Installing Motif 6

P516782.PDF [Page: 11 of 70]

III. LINUX DEVICE NAMES 7

IV. LINUX PROBLEMS 8
1. BOOTING 8
2. GNOME TERMINAL 8
3. MISCELLANEOUS PROBLEMS 8

V. INSTALLATION AND USE OF APPLICATION SOFTWARE 9
1. CHANG SOFTWARE 9
1.1 Copying the Entire Source Tree 9
1.2 Copying the Binaries 10
1.3 Machine Specific Settings 11
2. PRODAT 11
3. DISTRIBUTION TAR FILES 13

VI. INSTALLING PGPLOT 14

VII. SAPLOTX 16
1. VMS SAPLOTX 17
Running SAPLOTX 17
Building SAPLOTX 18
2. IRIX SAPLOTX 18
Running SAPLOTX 18
Building SAPLOTX 19
3. LINUX SAPLOTX 19
Running SAPLOTX 19
Building SAPLOTX 20
4. WINDOWS NT SAPLOTX 20
Running SAPLOTX 20
Building SAPLOTX 21

VIII. PROPWR 22
1. TESTING PROPWR 22
2. RUNNING PROPWR 22
3. BUILDING PROPWR 23

P516762.PDF [Page: 12 of 70]

IX. PORT OF VARIOUS FUNCTION LIBRARIES TO LINUX 27
1. TESTING THE LIBRARIES 27
2. DBIO LIBRARY 28
3. MTAPEIO LIBRARY 28
3.1 MFCOMMAND LIBRARY 29
4. MFCOMPRESS LIBRARY 29
5. MFLIB LIBRARY 29
6. WIDGETLIB LIBRARY 30
7. CONLIB LIBRARY 30
8. FPSMATH LIBRARY 30
9. SASIO LIBRARY 31

X. PORT OF VARIOUS PROGRAMS TO LINUX 32
1. TESTING THE PROGRAMS 32
2. DUMPDAT 32
3. DIRTAPE 32
4. DIRDISK 33
5. COPYDAT 33
6. TPULSE 34
7. DISPX 34
8. ASCDAT 35
9. GENCOM 35
10. PLTPWR 36
Compiling PLTPWR 36
PLTPWR Printing 37
Running PLTPWR 37
11. HETFIR 38
12. FIR6, OPTE AND FIRCOF 38

XI. PORT OF PRODAT TO LINUX 40
1. BACKGROUND 40
2. CVS REPOSITORY 40
PRODAT 40

P516762.PDF [Page: 13 of 70]

PRODAT Test Suite (PTS) 41
3. PORTING ISSUES 42
4. TESTING PRODAT 42
5. RUNNING PRODAT 43
Configuring X 43
PRODAT Printing 44
Text Interface to PRODAT 44
PRODAT GUI 44

XII. APPENDIX A - INSTALLING LINUX ON A WINDOWS 2000 DESKTOP SYSTEM 46
1. ATTEMPTING TO INSTALL LINUX ON THE WINDOWS DISK 46
2. ATTEMPTING TO INSTALL RED HAT LINUX 6.0 47
3. INSTALLING RED HAT LINUX 7.0 47
3.1 Preparing for Installation 47
3.2 Partitioning the Disks 48
3.3 Installing Linux 48

References 52

List of Tables

Table 1. Partitioning the second EIDE Drive 3
Table 2. Partitions on the first EIDE disk /tmp/hda (2482 cylinders): 4
Table 3. Partitions on the second EIDE disk /tmp/hdb (2482 cylinders): 4
Table 4. Linux device names. 7
Table 5. Analogs to the clean_linux and make_linux file for other platforms 10
Table 6. CTEST Programs and descriptions. 27
Table 7. CURSES Programs and descriptions. 28
Table 8. Partitioning of the second EIDE Drive on Windows 2000 machine 48
Table 9. Creating disk partitions for Linux 49

I. INTRODUCTION

The object of this contract was to install Linux and some application software on a Dell OptiPlex GX1 desktop computer with the following hardware:

1. Pentium III processor, 600 MHz.
2. 256 MB SDRAM memory.
3. 512 KB level 2 cache.
4. ATI 3D RAGE PRO TURBO, AGP 2X (GT-C2U2), video controller with 4 MB SGRAM.
5. Sound Blaster Pro compatible card (Crystal Ware audio driver CS32BA11).
6. 3Com Fast EtherLink XL 10/100 MB TX Ethernet network interface card.
7. Integrated EIDE controller.
8. Two 20 GB Maxtor 9 2049U6 EIDE hard disk drives.
9. IDE/ATAPI CD-ROM.
10. Adaptec AHA-2940UW (Ultra Wide) SCSI controller card.
11. Quantum DLT 4000 tape drive (SCSI ID 1).
12. 1.44 MB floppy disk drive.
13. PS/2 compatible mouse.
14. 104 key keyboard.
15. Philips 107B10 colour monitor.

The operating system was Microsoft Windows NT Workstation Version 4.0 (service pack 6). (As an auxiliary task, Red Hat Linux 7.0 was installed on a Windows 2000 system - see Appendix A.)

II. INSTALLING RED HAT LINUX 6.0

The installation followed the method used to put Red Hat Linux 6.0 on saturn.drea.dnd.ca (reference [7]). Rather than repeat that information here, we will only mention features peculiar to the thorne.drea.dnd.ca installation. In the following discussion, the "Official Red Hat Installation Guide" will be hereafter referred to as the "Installation Guide" or simply the "guide".

1. PREPARING FOR INSTALLATION

As it turned out, the installation program was able to automatically identify most of the hardware. The information that we had to provide was:

1. The mouse's type and number of buttons (generic PS/2 with two buttons).

2. Whether or not the disk drives were accessed in LBA (linear block addressing) mode, sometimes referred to as "large disk" mode. We were unable to determine if the disk drives used LBA mode. It was presumed that they did use LBA mode since they were new, large drives.

3. Network-related information:

IP address	
netmask	
gateway IP address (router)	
name server IP addresses	
domain name:	drea.dnd.ca
hostname:	thorne.drea.dnd.ca

4. Horizontal and vertical sync rates for the monitor. These were obtained from the web site http://www.pcstuff.philips.com/monitors/. Although the monitor did have a user's guide on a CD, that guide did not provide adequate technical specifications.

2. PARTITIONING THE DISKS

Prior to the Linux installation, the disk drives were partitioned as follows. The first EIDE drive was divided into three primary partitions: a 2047 MB FAT partition (MB=1,048,576 bytes), a 149 MB NTFS partition, and a 17273 MB NTFS partition. The Windows NT Workstation operating system was located in the FAT partition. The larger NTFS partition contained a Windows NT Server operating system which was not being used. The second EIDE drive contained a single 19469 MB NTFS partition used for temporary data storage.

We decided to make no changes to the first EIDE hard drive, and to partition the second hard drive as follows:

Table 1. Partitioning the second EIDE Drive

Type:	FAT root (/)	Linux swap	Linux swap	Linux swap	Linux /usr	Linux /home	Linux
Size (MB):	500	250	125	125	125	1000	Rest of disk.

The first two partitions are primary ones. The swap, /usr and /home partitions are logical partitions within an extended partition.

2.1 Performing the Partitioning

Prior to repartitioning the second disk, an NT repair floppy disk was created via the "RDISK /S" command. Although RDISK issued no warning messages, we found that there was too much repair information to store on a single floppy disk. Therefore we had to manually copy the remaining repair files from C:\WINNT\REPAIR to a second floppy disk.

The NT Disk Administrator program was used to delete the NTFS partition on the second hard drive and to create the FAT partition on that drive. Creation of the Linux partitions was done later via the fdisk program (see below).

3. INSTALLING LINUX

The installation process is generally straightforward, following the procedure starting on page 33 of the Installation Guide. Special steps for our particular installation are mentioned in the following paragraphs.

3.1 Creating Partitions for Linux (pp. 48 - 57 of guide)

The partitioning via fdisk proceeded as follows. For the second EIDE drive, we added a 250 MB Linux native primary partition (for /) after the existing FAT partition. Then we created an extended partition which occupied the rest of the disk. The extended partition was subdivided into five logical partitions. The first three logical partitions (Linux swap) occupied 125 MB each. The fourth logical partition was a 1000 MB Linux native partition for /usr. The fifth logical partition, a Linux native partition for /home, took up the rest of the extended partition.

After partitioning the disks, we copied down fdisk's partition tables for future reference. Following are the tables, plus the mount points we assigned later.

Table 2. Partitions on the first EIDE disk /tmp/hda (2482 cylinders).

Mount Point	Device	Boot	Start	End	System
	/tmp/hda1	*	1	261	DOS 16-bit >= 32M
	/tmp/hda2		262	280	OS/2 HPFS
	/tmp/hda3		281	2482	OS/2 HPFS

Table 3. Partitions on the second EIDE disk /tmp/hdb (2482 cylinders).

Mount Point	Device	Boot	Start	End	System
/fat1	/tmp/hdb1		1	64	DOS 16-bit >= 32M
/	/tmp/hdb2		65	96	Linux native
	/tmp/hdb3		97	2482	Extended
	/tmp/hdb5		97	112	Linux swap
	/tmp/hdb6		113	128	Linux swap
	/tmp/hdb7		129	144	Linux swap
/usr	/tmp/hdb8		145	272	Linux native
/home	/tmp/hdb9		273	2482	Linux native

fdisk had identified the NTFS partitions as being as OS/2 HPFS ones. This is typical behaviour on the part of Linux fdisk. (Note that Windows NT 4.0 does not support HPFS partitions.) The first Linux swap partition has mount point /tmp/hdb5, rather than /tmp/hdb4, since it is a logical partition. (Logical partitions are numbered from 5 upwards.)

3.2 Filesystem Configuration (p. 57 of guide)

The mount points were assigned as listed in the above table. We intentionally did not mount any partitions containing the Windows NT system files, to reduce the possibility that Linux users might damage these files. Any file sharing between Windows NT and Linux can be done via the /fat1 partition on the second hard drive.

3.3 Selecting Components (p. 61 of guide)

We chose components, rather than individual packages. In addition to the default components, we included the following:

1. Printer Support
2. KDE
3. DOS/Windows Connectivity

4. Graphics Manipulation
5. TeX Document Formatting
6. EMACS
7. EMACS with X Windows
8. C Development
9. Development Libraries
10. C++ Development
11. X Development
12. Gnome Development
13. Extra Documentation

KDE was included since KDE reportedly had fewer bugs than GNOME.

3.4 Package Installation (p. 64 of guide)

Formatting the drives, checking for bad blocks, and installing the packages took between 1 and 1.5 hours.

3.5 Configuring the X Window System (pp. 108-109 of guide)

Under "Monitor Setup", there were two entries for "Phillips (sic) 107B (17 inch/CM6800)". Since we did not know which of these applied to our monitor, we selected "custom" instead. The Red Hat Linux installation program does not allow one to arbitrarily specify the monitor's horizontal and vertical sync rates; instead one must select from a menu of rates. For the horizontal sync rate we chose "1600 x 1200 @ 70 Hz", and for the vertical sync rate we picked "50-100". We judged that these were the closest settings corresponding to one of the monitor's preset resolution modes (1600 X 1200 @ 70 Hz), and to the monitor's vertical scanning range (50 to 160 Hz).

4. LINUX CONFIGURATION

The configuration generally followed the procedure used for saturn.drea.dnd.ca (reference [7]). Rather than repeat that discussion here, we will only mention features peculiar to the thorne.drea.dnd.ca installation. Note that IDL was not installed on thorne.

4.1 Allowing an Ordinary User to Mount the Floppy Disk

Unlike in the saturn.drea.dnd.ca configuration, an ordinary user was able to mount the floppy disk. No additional work was required to make this possible.

4.2 Allowing an Ordinary User to Access the Tape

If the tape drive was turned on before Linux was booted, the following message would appear during the boot process: "Can't locate module st". The solution to this problem is given in the Linux FAQ ("Linux Frequently Asked Questions and Answers") in the file

/usr/DOC/FAQ/txt/Linux-FAQ. As per section 9.1 of the FAQ ("Modprobe can't locate module 'XXX' and similar messages"), we added the following line to the file /etc/conf.modules: "alias st off". Subsequently, the error message did not appear during the Linux boot.

To permit an ordinary user to access the tape, we changed the permissions for some files in the /dev directory:

```
chmod o+rw /dev/st0
chmod o+rw /dev/nst0
```

4.3 Allowing Remote FTP Access to the Computer

Installation of an FTP server (wu-ftpd package) was done by running GnoRPM.

4.4 Configuring a Printer

This was done by following the instructions given of pages 158 to 164 of the Red Hat Installation Guide. We planned to use the remote HP 4050 Laser Jet PostScript printer in room 335 at DREA. From the manufacturer's web site (http://www.hp.com), we found that the printer had a 1200 dpi resolution.

During the Linux printer configuration, for "Printer Type" we chose "Remote Unix (lpd) Queue". We specified the remote host as "hp-335.drea.dnd.ca" and left the "Remote Queue" field blank. For "Input Filter" we selected "PostScript Printer" with 1200 X 1200 resolution and letter size paper.

4.5 Installing Motif

Motif is a GUI development library which is used by the PRODAT Front End (PFE). Two Motif implementations available for Linux are Lesstif (a clone of Motif 1.2; see http://www.lesstif.org) and Open Motif (Motif 2.1 for Open Source operating systems; see http://www.opengroup.org/openmotif). An old version of Lesstif (version 0.86.9-1 as used by the IBM application Via Voice) was present on the Red Hat Linux 6.0 Applications CD. However, we decided to install Open Motif, since Lesstif is incompatible with some Motif applications. The runtime and development Linux binary packages for Open Motif (version 2.1.30-4) were downloaded from http://www.metrolink.com/openmotif and installed. (The RPM packages are currently in the directory /root/openmotif on thorne.)

III. LINUX DEVICE NAMES

Some of this information can be obtained by examining the file /etc/fstab.

Table 4. Linux device names.

Device	NT Drive	Linux Device	Linux Mount Point	Type	Mounted at Linux Boot?
Floppy disk	A:	/dev/fd0	/mnt/floppy	ext2	No
1st EIDE disk	C:	/dev/hda1		vfat	No
	D:	/dev/hda2		ntfs	No
	E:	/dev/hda3		ntfs	No
2nd EIDE disk	F:	/dev/hdb1	/fat1	vfat	Yes
		/dev/hdb2	/	ext2	Yes
		/dev/hdb8	/usr	ext2	Yes
		/dev/hda9	/home	ext2	Yes
CD-ROM	G:	/dev/cdrom	/mnt/cdrom	iso9660	No
SCSI tape	Tape1	/dev/st0 /dev/nst0 etc.			
Sound card		/dev/audio /dev/dsp etc.			

Notes:

1. "Type" is the default file system type. It can be changed via the "-t" parameter of the "mount" command.

2. For further information regarding the SCSI tape device, refer to the man page for "st".

3. For further information regarding sound cards, refer to the file /usr/doc/HOWTO/Sound-HOWTO.

IV. LINUX PROBLEMS

1. BOOTING

At one point when we attempted to boot Linux via the Windows NT loader, "LIL" appeared on the screen, rather than the expected "LILO boot:" prompt, and Linux did not boot. It appeared that the boot sector on /tmp/hdb2 had been changed. Fortunately, we were still able to boot Linux by using the Linux boot floppy.

To remedy the problem, we copied the revised Linux boot sector to the Windows NT file C:\BOOTSECT.LNX. Subsequently, the Windows NT loader was able to successfully boot Linux.

2. GNOME TERMINAL

Telnetting to the DREA FIRM cluster from a Gnome terminal sometimes caused problems; the Gnome terminal would keep overwriting the same line on the screen. The only way to clear this condition was to close the Gnome terminal. Such problems do not exist if one instead uses an xterm.

3. MISCELLANEOUS PROBLEMS

a. On two occasions, the login session bombed with a core dump while a file was being edited with the vi editor (vim).

b. On one occasion, the Gnome desktop took an inordinate amount of time (several minutes) to appear after login. Subsequently, the system did not respond to the "logout" item in the Gnome main menu; instead one had to enter the "shutdown" command via a Gnome terminal to get out of the Linux session.

V. INSTALLATION AND USE OF APPLICATION SOFTWARE

The following sections describe how to compile and run the signal processing software that was ported to Linux. The software has been split into two camps: PRODAT (in the Chang2 directory), and all the other software (in the Chang directory). The reason for this division is that PRODAT is normally maintained by other people.

Note that the application software was ported to Red Hat Linux 6.0 on an Intel machine. There is no guarantee that it will work under other Linux distributions or on other machines.

1. CHANG SOFTWARE

1.1 Copying the Entire Source Tree

To transfer the software to your own Linux machine, tar the appropriate parts of the Chang directory on thorne, and then copy the tar file to your machine. (All required Chang files and directories permit read access by all users.) Assume that user "guest" wishes to create such a tar file; this can be done as follows:

```
cd /home/chang
tar -cvf /home/guest/chang.tar --exclude="Desktop/*" [A-Zcm]*
```

Due to binary incompatibilities between Linux systems, it may be necessary to recompile the Chang software on your computer. To aid in rebuilding the software, two script files have been provided in the Chang home directory: clean_linux to remove the existing binaries, and make_linux to recompile the software. Some changes to these scripts may be necessary, depending on where you have installed the Chang software. These scripts are used as follows:

```
clean_linux [-a]
make_linux [-a]
```

The optional "-a" flag indicates that you wish to include the PGPLOT binaries in the clean/make operation. Since rebuilding the PGPLOT binaries is a complex operation, one may wish to do this separately (see below). If so, one should build the PGPLOT files before running make_linux.

Analogs to the clean_linux and make_linux files exist in the Chang home directories on other platforms:

Table 5. Analogs to the clean_linux and make_linux file for other platforms.

Platform	Clean Script	Make Script
Linux	clean_linux	make_linux
IRIX	clean_sgi	make_sgi
Windows NT	clean_nt.bat	make_nt.bat
VMS	clean_vms.com	make_vms.com

1.2 Copying the Binaries

Rather than copy the entire Chang source tree to another machine, one may wish to copy only the binary executables for some or all of the programs. If so, one should note the following dependencies:

1. Programs that use the PGPLOT library need the following files from the /home/chang/PGPLOT/pgplot directory: grfont.dat, rgb.txt, and pgxwin_server.
2. Program PLTPWR needs the following files from the /home/chang/PLTPWR directory: pltpwr.help and lpr_post.
3. Programs TPULSE and TBUOY need the decim08.dat and decim32.dat data files from the /home/chang/SONAR directory.

The script file copy_linux in the /home/chang directory can be used to copy all the Chang binary executables and data files, or to provide links to those files. (Your current working directory must be the /home/chang directory, or copy thereof, to run this script.) This script is used as follows:

copy_linux [-s] Destination_dir

The optional "-s" flag indicates that one wishes to create symbolic links to the Chang files, rather than copy them. "Destination_dir" is the directory where the Chang files (or links to them) will be copied.

Once one has set up the files in Destination_dir, one can modify one's login file to permit ready use of those files. (To determine your login shell, enter the command "printenv SHELL". One can change one's login shell via the chsh command.)

The following examples show how one might modify one's login file if Destination_dir is /usr/local/bin. If Destination_dir is already in your PATH, do not add the last line which changes PATH. (To see if Destination_dir is already in your PATH, use the command "printenv PATH".) If your login shell is bash, add the following lines to the end of the .bash_profile file in your home directory:

```
export PGPLOT_DIR=/usr/local/bin/PGPLOT/
export PLTPWR_HELP=/usr/local/bin/pltpwr.help
export PROPWR_HELP=/usr/local/bin/propwr.help
export FILTERS_DIR=/usr/local/bin/
export PATH=$PATH:/usr/local/bin
```

If your login shell is csh or tcsh, add the following lines to the end of the .login file in your home directory:

```
setenv PGPLOT_DIR /usr/local/bin/PGPLOT/
setenv PLTPWR_HELP /usr/local/bin/pltpwr.help
setenv PROPWR_HELP /usr/local/bin/propwr.help
setenv FILTERS_DIR /usr/local/bin/
setenv PATH ${PATH}:/usr/local/bin
```

By including Destination_dir in one's path, one can execute say copydat by entering:

```
copydat
```

as opposed to something like:

```
/home/chang/SONAR/copydat or /usr/local/bin/copydat
```

1.3 Machine Specific Settings

The programs require that certain environment variables be set appropriately (TERM, TERM_TYPE, FILTERS_DIR, PLTPWR_HELP, PROPWR_HELP, PGPLOT_DIR, and PGPLOT_XW_WIDTH). Moreover, the script file lpr_post should be customized for one's own machine. See below for more information on these settings.

2. PRODAT

Files related to PRODAT are stored in the directory /home/chang2 on thorne. Three subdirectories in this directory are:

1. CVSROOT: This is contains a CVS source code repository based on Chris Maxwell's repository (as of January 2001) on the SGI warrior computer. The repository has been modified to include the PRODAT LINUX_X86 branch release for the Linux Intel X86 platform, other Linux related changes, and bug fixes.
2. Linux_Prodat: This contains the source and binary code for PRODAT.
3. Prodat_Testing: This contains files related to the PRODAT Test Suite (PTS).

The subdirectory of interest to the general user is Linux_Prodat. In the following discussion, a "link" is a means of accessing a file by more than one name (see "man ln"). The PRODAT

Front End (PFE) is a program which provides a GUI front end to PRODAT. The SDU Server is a program which acts as an intermediary between PRODAT and the X Window system when plotting data.

Important files in the Linux_Prodat subdirectory are:

1. In Linux_Prodat/src/Prodat:
    a. lpr_post - A symbolic link to prodat/lpr_post, a script for printing PostScript files.
    b. prodat.sh - A script to check for orphaned SDU Servers and to run PRODAT. The variable PRODAT_PATH in this script should be modified to match one's particular set-up. (*)
    c. Prodat - A symbolic link to prodat/prodat, the PRODAT program.
    d. SDU_Server - A symbolic link to sdu3_0/SDU_Server, the SDU Server program.

2. In Linux_Prodat/src/Prodat/pfe:
    a. pfe - The PRODAT Front End program. (*)
    b. run_prodat - A symbolic link to ../prodat.sh, a script for running PRODAT. The output of pfe is passed to the input of run_prodat.

3. In Linux_Prodat/src/Prodat/prodat:
    a. lpr_post - A script for printing PostScript files. (*)
    b. prodat - The PRODAT program.

4. In Linux_Prodat/src/Prodat/sdu3_0:
    a. SDU_Server - The SDU Server program. (*)

Assuming that one's system can run Red Hat Linux 6.0 binaries, one can just copy the PRODAT, SDU Server, and PFE programs plus the prodat.sh and lpr_post scripts to one's own system. (See items marked with an asterisk (*) above.) The scripts will probably need local modifications, and one will have to create appropriate link files as noted above.

The script file copypro_linux in the /home/chang2 directory can be used to copy all the PRODAT binary executables and script files, or to provide links to those files. (Your current working directory must be the /home/chang2 directory, or copy thereof, to run this script.) This script is used as follows:

copypro_linux [-s] Destination_dir

The optional "-s" flag indicates that one wishes to create symbolic links to the PRODAT files, rather than copy them. "Destination_dir" is the directory where the PRODAT files (or links to them) will be copied.

If Destination_dir is not already in your PATH, you can add a line to your login file to put it in your path. In the following examples, it is assumed that Destination_dir is /usr/local/bin. If your login shell is bash, add the following line to the end of the .bash_profile file in your home directory:

```
export PATH=$PATH:/usr/local/bin
```

If your login shell is csh or tcsh, add the following line to the end of the .login file in your home directory:

```
setenv PATH ${PATH}:/usr/local/bin
```

Should one be unable to run the PRODAT binaries, it will be necessary to recompile the programs on your system. One way of obtaining the PRODAT source code would be to tar the contents of the /home/chang2/Linux_Prodat directory and transfer that tar file to your system. User "guest" could create such a tar file as follows:

```
cd /home/chang2
tar -cvf /home/guest/prodat.tar Linux_Prodat copypro_linux
```

Note that it is necessary to have Motif library on your system to compile the PRODAT Front End (see section II.4.5 above). To recompile the PRODAT software, issue the following commands in the Linux_Prodat/src/Prodat directory:

```
make clean
make all
```

As noted above, it will probably be necessary to modify the prodat.sh and lpr_post scripts to suit one's own set-up.

3. DISTRIBUTION TAR FILES

Some distribution tar files are currently stored in the /home/guest directory on the thorne computer. In the following file names, "ddmmmyy" is the date on which the file was created.

a. chang_bin_ddmmmyy.tar.gz - The Chang binary files.
b. chang_src_ddmmmyy.tar.gz - The Chang source and binary files.
c. prodat_bin_ddmmmyy.tar.gz - The PRODAT binary files.
d. prodat_src_ddmmmyy.tar.gz - The PRODAT source and binary files.

P516762.PDF [Page: 28 of 70]

VI. INSTALLING PGPLOT

PGPLOT is a device-independent graphics package for making simple scientific graphs (http://astro.caltech.edu/~tjp/pgplot). The package is used by several DREA programs. The distribution file for PGPLOT is in the following file on the DREA SGI computer warrior:

/usr/people/chang/PGPLOT/src/pgplot5.2.tar.gz

Instructions for installing PGPLOT on UNIX systems can be found in the following warrior file:

/usr/people/chang/PGPLOT/src/pgplot/install-unix.txt

There are three compiler options for installing PGPLOT under Linux. For this particular installation, the option used was "g77_elf". Further information regarding ELF can be obtained from the Linux file /usr/DOC/FAQ/txt/Linux-FAQ (section 8.2 - "What's all this about ELF?").

Both static (libpgplot.a) and shared (libpgplot.so) PGPLOT libraries were created in the thorne directory /home/chang/PGPLOT/pgplot. Another library in this directory (libcpgplot.a) provides a C binding to PGPLOT. Also in this directory is a list of the device drivers installed (drivers.list).

A problem was encountered when building the PGPLOT demonstration programs, since reference was made to the non-existent directory /usr/X11. This problem was overcome by issuing the following command in the /usr directory:

ln -s X11R6 X11

The PGPLOT Linux installation was tested by running the demonstration programs present in /home/chang/PGPLOT/pgplot. Before running any PGPLOT program, one must set the environment variable PGPLOT_DIR which specifies the directory where the files grfont.dat and rgb.txt can be found. This can be done under the bash shell via:

export PGPLOT_DIR=/home/chang/PGPLOT/pgplot/

Since the demonstration programs used the shared PGPLOT library, one had to put the PGPLOT directory in the loader link path. This can be done under bash via:

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/chang/PGPLOT/pgplot

Testing PGPLOT with a remote /TEK4010 device was successful. However, testing PGPLOT with a local /XSERVE device (persistent window on X Window System) revealed the following shortcomings:

a. Adjusting contrast, brightness and palettes did not work (pgdemo4).

b. Moving a rubber-band line very quickly would leave behind "vapor trails" on the screen (pgdemo5).

These shortcomings are due to limitations of the hardware and of Linux, and are not unexpected in PC X terminal emulations.

P516762.PDF [Page: 30 of 70]

VII. SAPLOTX

SAPLOT (the DREA Surveillance Acoustics Plotting Package) is a graphics package used to produce publication quality plots from text files containing plot commands and numeric plot data (reference [14]). SAPLOT_PS is a version of SAPLOT which generates PostScript files for output, and allows previewing of graphics on a Tektronix 40XX compatible terminal or emulator. Documentation for SAPLOT_PS can be found in the file SAPLOT_PS.DOC in the DREA FIRM directory SOFTPACS:[VEST.SAPLOT_PS]. To facilitate porting of SAPLOT_PS to various platforms, SAPLOT_PS was modified so that its preview functions would be done via the PGPLOT library (see above). This modified version of SAPLOT_PS is known as SAPLOTX.

SAPLOTX has mostly the same features and limitations as SAPLOT_PS (e.g., it cannot be used to preview grey scale plots). Some exceptions are:

1. In SAPLOTX the command line plot preview switch can be "/s", "/S", "-s" or "-S" on any platform.
2. SAPLOTX supports more terminal types than SAPLOT_PS. Colour plots are possible, provided that the terminal has colour capabilities. Note that the colours used will not always match those specified by the SAPLOTX commands. For instance, PGPLOT uses a black background for colour terminal output, whereas SAPLOTX specifies a white background for its plots.
3. The environment variable used to specify the terminal type for SAPLOTX is "TERM_TYPE" rather than "TERM$TYPE". (The "$" character has special meaning in most UNIX shells.)
4. In SAPLOTX, the default terminal type is "MSWIN" under Microsoft Windows, and "XWIN" on other platforms.

Terminal types recognized by SAPLOTX are:

KERMIT:	Kermit version 3 on the IBM PC
MSWIN:	Microsoft Windows (default on Microsoft systems)
NCSA:	NCSA Telnet on the Macintosh
NCSAMAC:	NCSA Telnet on the Macintosh
TEK4010:	Tektronix 4010
TEK4100:	Tektronix 4100
VT100:	vt240 (also vt330) in vt100 mode
VT240:	vt240 (also vt330) in vt200 mode
VT640:	vt100 with Retrographics
VTMAC:	VersaTerm-PRO (Tek 4014) on the Macintosh
VTMAC4105:	Versaterm-PRO (Tek 4105) on the Macintosh
XTERM:	Xterm
XWIN:	X window (default on non-Microsoft systems)

The terminal type "MSWIN" can only be used on Microsoft systems, and it is the only

allowable terminal type there.

SAPLOTX is available for the following operating systems: VMS, SGI IRIX, Linux, and Windows NT. Before running SAPLOTX, one must define several environment variables. The user may wish to define these variables in their login files (e.g., the bash .bash_profile file or the csh .login file).

1. VMS SAPLOTX

Running SAPLOTX

SAPLOTX is available on the DREA FIRM cluster. Before running SAPLOTX, the logical names PGPLOT_DIR and GRPSHR must be defined. This can be done via the following DCL command:

```
@PS_72:[CHANG.PGPLOT.AXP]DREA_LOGICAL
```

One can optionally define the PGPLOT X window width via the following DCL command:

```
DEFINE PGPLOT_XW_WIDTH number
```

where "number" is a fraction (<= 1.0) of the screen width. This command should be entered before the program starts the PGPLOT X window server pgxwin. If pgxwin is already running, the user should close it before defining PGPLOT_XW_WIDTH.

To specify the terminal type for SAPLOTX, use the DCL command:

```
DEFINE TERM_TYPE atype
```

where "atype" is the SAPLOTX terminal type as selected from the list given above (e.g., DEFINE TERM_TYPE TEK4010).

SAPLOTX must be installed as a VMS foreign command in order to pass command line parameters to it. The installation can be done via the following DCL command:

```
SAPLOTX == "$ PS_72:[CHANG.SAPLOTX.AXP]SAPLOTX"
```

The '$' and the device name (currently PS_72) are required.

SAPLOTX can be run as follows:

```
SAPLOTX [ input_file [ output_file ] ] [/S]
```

where "input_file" is the input SAPLOTX data file, "output_file" is the output PostScript file, and "/S" (alternatively "/s", "-S" or "-s") is a switch indicating that one wishes to preview the plot. Note that if "/S" is present, no PostScript file is produced.

Building SAPLOTX

One must first define PGPLOT_DIR as mentioned in the previous section. To compile and link SAPLOTX, one then enters the following DCL command in the PS_72:[CHANG.SAPLOTX] directory:

```
@SAPMAKE
```

2. IRIX SAPLOTX

Running SAPLOTX

SAPLOTX is available on the DREA SGI computers unicorn and warrior. Before running SAPLOTX, the environment variable PGPLOT_DIR must be defined as per the following command(s):

csh or tcsh:

```
setenv PGPLOT_DIR /usr/people/chang/PGPLOT/pgplot/
```

sh:

```
PGPLOT_DIR="/usr/people/chang/PGPLOT/pgplot/"; export PGPLOT_DIR
```

One can optionally define the PGPLOT X window width via the following command(s):

csh or tcsh:

```
setenv PGPLOT_XW_WIDTH number
```

sh:

```
PGPLOT_XW_WIDTH="number"; export PGPLOT_XW_WIDTH
```

where "number" is a fraction (<= 1.0) of the screen width. This command should be entered before the program starts the PGPLOT X window server pgxwin. If pgxwin is already running, the user should close it before defining PGPLOT_XW_WIDTH.

To specify the terminal type for SAPLOTX, use the following command(s):

csh or tcsh:

```
setenv TERM_TYPE atype
```

sh:

```
TERM_TYPE="atype"; export TERM_TYPE
```

where "atype" is the SAPLOTX terminal type as selected from the list given above (e.g., setenv TERM_TYPE TEK4010).

SAPLOTX can be run as follows:

/usr/people/chang/SAPLOTX/saplotx [input_file [output_file]] [/S]

where "input_file" is the input SAPLOTX data file, "output_file" is the output PostScript file, and "/S" (alternatively "/s", "-S" or "-s") is a switch indicating that one wishes to preview the plot. Note that if "/S" is present, no PostScript file is produced.

Building SAPLOTX

One must first define PGPLOT_DIR as mentioned in the previous section. To compile and link SAPLOTX, one then enters the following command in the /usr/people/chang directory:

make sgi

3. LINUX SAPLOTX

Running SAPLOTX

SAPLOTX is available on the DREA computer thorne. Before running SAPLOTX, the environment variable PGPLOT_DIR must be defined as per the following command:

csh or tcsh:

setenv PGPLOT_DIR /home/chang/PGPLOT/pgplot/

bash:

export PGPLOT_DIR=/home/chang/PGPLOT/pgplot/

One can optionally define the PGPLOT X window width via the following command:

csh or tcsh:

setenv PGPLOT_XW_WIDTH number

bash:

export PGPLOT_XW_WIDTH=number

where "number" is a fraction (<= 1.0) of the screen width. This command should be entered before the program starts the PGPLOT X window server pgxwin. If pgxwin is already

running, the user should close it before defining PGPLOT_XW_WIDTH.

To specify the terminal type for SAPLOTX, use the following command:

csh or tcsh:

setenv TERM_TYPE atype

bash:

export TERM_TYPE=atype

where "atype" is the SAPLOTX terminal type as selected from the list given above (e.g., setenv TERM_TYPE TEK4010).

SAPLOTX can be run as follows:

/home/chang/SAPLOTX/saplotx [input_file [output_file]] [/S]

where "input_file" is the input SAPLOTX data file, "output_file" is the output PostScript file, and "/S" (alternatively "/s", "-S" or "-s") is a switch indicating that one wishes to preview the plot. Note that if "/S" is present, no PostScript file is produced.

Building SAPLOTX

One must first define PGPLOT_DIR as mentioned in the previous section. To compile and link SAPLOTX, one then enters the following command in the /home/chang/SAPLOTX directory:

make linux

When building SAPLOTX, one will get the following error message: "the 'gets' function is dangerous and should not be used". (For further details, see the man page for "gets".) Since it is unlikely that the use of "gets" in SAPLOTX will result in a buffer overflow, we have not bothered to replace "gets" with "fgets".

4. WINDOWS NT SAPLOTX

Running SAPLOTX

SAPLOTX is available on the DREA computer auk. Before running SAPLOTX, the environment variable PGPLOT_DIR must be defined as per the following command:

SET PGPLOT_DIR=D:\Chang\PGPLOT\NT\

The default on-screen graphic window size is 800*600. The default can be changed by setting the environment variable PGPLOT_VIDEO as follows:

SET PGPLOT_VIDEO=VGA (or V), 640 * 480
SVGA (or S), 800 * 600
XGA (or X), 1024 * 768
ZGA (or Z), 1280 * 1024

Modes exceeding the capability of the Windows screen driver will be reduced to the maximum available.

SAPLOTX can be run as follows:

D:\Chang\SAPLOTX\SAPLOTX [input_file [output_file]] [/S]

where "input_file" is the input SAPLOTX data file, "output_file" is the output PostScript file, and "/S" (alternatively "/s", "-S" or "-s") is a switch indicating that one wishes to preview the plot. Note that if "/S" is present, no PostScript file is produced.

Building SAPLOTX

One must first define PGPLOT_DIR as mentioned in the previous section. To compile and link SAPLOTX, one then enters the following command in the D:\Chang\SAPLOTX directory:

NMAKE /F MAKEFILE_NT ALL

The Windows NT version of SAPLOTX differs from other versions in that its main program is written in Fortran rather than C. This was necessary to get SAPLOTX to work with the NT version of PGPLOT.

VIII. PROPWR

PROPWR is a program for processing power spectrum files (references [12] and [10]). PROPWR was changed so that it could run on the IRIX, Linux, and Windows NT platforms, as well as VMS.

1. TESTING PROPWR

The PROPWR code was run through a Fortran static code checker (FTNCHEK) in order to detect possible problems. Run time testing of PROPWR was done by ensuring that its output on the IRIX, Linux, and Windows NT platforms agreed (within calculation accuracy) with its output on the VMS platform. As a result of the above tests, several bugs were found and corrected in PROPWR:

a. Some undefined variables were being used.
b. There was a conflict between the unit numbers used by some SASIO files (e.g., the macro files) and the data files generated by the program.
c. The byte and ASCII blocks of the input power spectrum file were not stored properly in the output tracked data (.SVT) file.
d. If one specified a new name for the output tracked data (.SVT) file, the new file name did not appear in the screen display.
e. If one turned off some types of processing (e.g., "TRACK OFF"), the associated output file was still shown in the screen display.
f. Regarding the TRACK -> GSET command: If one specified some tracking bands and immediately switched to zoom mode, the bands just entered in non-zoom mode would be lost.

It was believed that there might be problems with the SAXA and OTA normalizers used by the program, since their noise estimates contained "flat" areas. However it was determined that such behaviour was correct for the given test data.

2. RUNNING PROPWR

PROPWR, like SAPLOTX, uses the PGPLOT graphics library. Therefore, the environment variable PGPLOT_DIR must be defined before running PROPWR. Moreover, the environment variable TERM must be defined on the IRIX and Linux platforms. See the above SAPLOTX section VII. for information about these environment variables.

Another environment variable which one may need to define is PROPWR_HELP, the name of the PROPWR help file. In the following examples, PROPWR_HELP is set to its default value.

VMS PROPWR

To run PROPWR on the DREA FIRM cluster enter:

```
DEFINE PROPWR_HELP PS_72:[CHANG.PROPWR]PROPWR.HELP
RUN PS_72:[CHANG.PROPWR.AXP]PROPWR
```

IRIX PROPWR

To run PROPWR on the SGI computers unicorn and warrior (csh or tcsh shell) enter:

```
setenv PROPWR_HELP=/usr/people/chang/PROPWR/propwr.help
/usr/people/chang/PROPWR/propwr
```

Linux PROPWR

To run PROPWR on the Linux computer thorne (bash shell) enter:

```
export PROPWR_HELP=/home/chang/PROPWR/propwr.help
/home/chang/PROPWR/propwr
```

Windows NT PROPWR

To run PROPWR on the Windows NT computer auk enter:

```
SET PROPWR_HELP=E:\Chang\PROPWR\propwr.help
E:\Chang\PROPWR\propwr
```

3. BUILDING PROPWR

VMS PROPWR

Program PROPWR requires the creation of the SASIO command table subroutine PROPWR_VMS_TAB.F. This can be done by entering the following commands while in the PS_72:[CHANG.PROPWR] directory on the FIRM cluster:

```
RUN PS_72:[CHANG.SASIO]GENCOM
INFIL PROPWR_VMS.TAB
TERM OFF
OFIL PROPWR_VMS_TAB.F
RUN
QUIT
```

Some editing must be applied to the resultant PROPWR_VMS_TAB.F file:

The statement

```
HLPFIL='PS_72:[CHANG.PROPWR]PROPWR.HELP'
```

should be replaced by

```
CALL SASENV('PROPWR_HELP',HLPFIL,LENHLP)
IF (LENHLP.LE.0) HLPFIL='PS_72:[CHANG.PROPWR]PROPWR.HELP'
```

PROPWR can be compiled and linked by entering the following commands from within the [CHANG.PROPWR] directory:

```
@PS_72:[CHANG.PGPLOT.AXP]DREA_LOGICAL
@MAKE_PROPWR
```

IRIX PROPWR

Program PROPWR requires the creation of the SASIO command table subroutine propwr_sgi_tab.f. This can be done by entering the following csh or tcsh shell commands while in the /usr/people/chang/PROPWR directory:

```
setenv TERM vt100 (Needed only if TERM isn't defined)
../SASIO/gencom
INFIL propwr_sgi.tab
TERM OFF
OFIL propwr_sgi_tab.f
RUN
QUIT
```

Some editing must be applied to the resultant propwr_sgi_tab.f file:

The statement

```
HLPFIL='/usr/people/chang/PROPWR/propwr.help'
```

should be replaced by

```
CALL SASENV('PROPWR_HELP',HLPFIL,LENHLP)
IF (LENHLP.LE.0) HLPFIL='/usr/people/chang/PROPWR/propwr.help'
```

PROPWR can be compiled and linked by entering the following commands from within the chang/PROPWR directory:

sh shell:

```
PGPLOT_DIR="/usr/people/chang/PGPLOT/pgplot/"
export PGPLOT_DIR
make sgi
```

P516762.PDF [Page: 39 of 70]

csh or tcsh shell:

```
setenv PGPLOT_DIR /usr/people/chang/PGPLOT/pgplot/
make sgi
```

Linux PROPWR

Program PROPWR requires the creation of the SASIO command table subroutine propwr_linux_tab.f. This can be done by entering the following bash shell commands while in the /home/chang/PROPWR directory on thorne:

```
export TERM=vt100 (Needed only if TERM isn't defined)
../SASIO/gencom
INFIL propwr_linux.tab
TERM OFF
OFIL propwr_linux_tab.f
RUN
QUIT
```

Some editing must be applied to the resultant propwr_linux_tab.f file:

The statement

```
HLPFIL='/home/chang/PROPWR/propwr.help'
```

should be replaced by

```
CALL SASENV('PROPWR_HELP',HLPFIL,LENHLP)
IF (LENHLP.LE.0) HLPFIL='/home/chang/PROPWR/propwr.help'
```

PROPWR can be compiled and linked by entering the following commands from within the chang/PROPWR directory:

bash shell:

```
export PGPLOT_DIR=/home/chang/PGPLOT/pgplot/
make linux
```

csh or tcsh shell:

```
setenv PGPLOT_DIR /home/chang/PGPLOT/pgplot/
make linux
```

P516762.PDF [Page: 40 of 70]

Windows NT PROPWR

Program PROPWR requires the creation of the SASIO command table subroutine propwr_nt_tab.f. This can be done by entering the following commands while in the PROPWR directory E:\Chang\PROPWR:

```
\Chang\SASIO\gencom
INFIL propwr_nt.tab
TERM OFF
OFIL propwr_nt_tab.f
RUN
QUIT
```

Some editing must be applied to the resultant propwr_nt_tab.f file:

The statement

```
INCLUDE 'prog_banner.f'
```

should be replaced by

```
INCLUDE 'prog_banner_nt.f'
```

The statement

```
HLPFIL='\Chang\PROPWR\propwr.help'
```

should be replaced by

```
CALL SASENV('PROPWR_HELP',HLPFIL,LENHLP)
IF (LENHLP.LE.0) HLPFIL='\Chang\PROPWR\propwr.help'
```

The Windows NT executable "propwr.exe" is created by using the file makefile_nt in E:\Chang\PROPWR thus:

```
SET PGPLOT_DIR=E:\Chang\PGPLOT\NT\
NMAKE /F makefile_nt all
```

IX. PORT OF VARIOUS FUNCTION LIBRARIES TO LINUX

Note that this section covers only the libraries that are present in the Chang directories on the VMS, SGI or NT platforms. It does not cover PRODAT's libraries, which are normally maintained by other people. The port of PRODAT to Linux is described below in a separate section.

Several of the Chang libraries are located in the directory /home/chang/MFLIB on the Linux computer thorne. Rather than building each library separately as described below, one can build all the libraries in this directory via:

make linux

1. TESTING THE LIBRARIES

The thorne directories /home/chang/CTEST and /home/chang/CURSES contain several programs for testing the Linux libraries. These test programs are built by issuing the following command in the CTEST and CURSES directories:

make linux

CTEST Programs

Table 6. CTEST Programs and descriptions.

Name	Description
check_ansilab	Examines ANSI labels on a tape (MFLIB library).
comprs	Tests COMPLEX*4 data compression routines (MFCOMPRESS library).
comprs2	Tests REAL*2 data compression routines (MFCOMPRESS library).
test_dbio	Tests the DBIO library functions.
test_mfswap	Tests data conversion routines (MFLIB library).
test_mtapeio	Tests the MTAPEIO library functions.
test_mftermin	Tests terminal I/O functions (MFLIB library).
test_mftermout	Tests terminal and report file I/O (MFLIB library).

When building program test_mfswap, one will get the following message: "warning decimal constant is so large that it is unsigned". The C compiler is objecting to the legitimate integer constant -2147483648. This warning can be ignored since the constant is handled properly by the compiler.

CURSES Programs

Table 7. CURSES Programs and descriptions.

Name	Description
curdemo1	Tests the curses library.
widgtest	Tests the WIDGETLIB library.

2. DBIO LIBRARY

DBIO is a library of functions for doing direct access disk block I/O. The VMS and IRIX versions of DBIO are described in reference [15]. The Windows NT version of DBIO is described in reference [2]. The Linux version of DBIO (DBIO_UNIX) is the same version used on the IRIX and NT platforms. However under Linux, DBIO_UNIX does not support asynchronous I/O.

On the Linux computer thorne, the source code for DBIO_UNIX is in the directory /home/chang/MFLIB. The DBIO library dbio_unix.a is created by using the file makefile_linux in this directory thus:

```
make -f makefile_linux dbio_unix.a
```

3. MTAPEIO LIBRARY

MTAPEIO is a library of functions for performing magnetic tape operations. The VMS and IRIX versions of MTAPEIO are described in reference [15]. The Windows NT version of MTAPEIO is described in reference [2]. The IRIX version of MTAPEIO (MTAPEIO_UNIX) was modified to permit its use under either IRIX or Linux. Note that the Linux version of MTAPEIO does support asynchronous I/O.

It was found that there was a bug in the Red Hat Linux 6.0 system tape driver. If one tries to skip backwards past the beginning of the tape using a skip file command, Linux will signal an end of file (EOF) condition rather than a beginning of tape (BOT) condition. Subsequently, Linux will always report a BOT condition as an EOF condition, unless one performs a rewind operation. MTAPEIO_UNIX takes steps to compensate for this bug.

On the Linux computer thorne, the source code for MTAPEIO_UNIX is in the directory /home/chang/MFLIB. The MTAPEIO library mtapeio_unix.a is created by using the file makefile_linux in this directory thus:

```
make -f makefile_linux mtapeio_unix.a
```

3.1 MFCOMMAND LIBRARY

The MFCOMMAND library contains some basic functions for implementing a command language user interface. The VMS and IRIX versions of MFCOMMAND are described in reference [15]. The Windows NT version of MFCOMMAND is described in reference [2].

On the Linux computer thorne, the source code for MFCOMMAND is in the directory /home/chang/MFLIB. The MFCOMMAND library mfcommand.a is created by using the file makefile_linux in this directory thus:

```
make -f makefile_linux mfcommand.a
```

4. MFCOMPRESS LIBRARY

MFCOMPRESS is a library of functions for compressing binary numbers. The VMS version of MFCOMPRESS is described in reference [8]. The IRIX and Windows NT versions of MFCOMPRESS are described in reference [2].

On the Linux computer thorne, the source files for MFCOMPRESS are in the directory /home/chang/MFLIB. The MFCOMPRESS library mfcompress.a is created by using the file makefile_linux in this directory thus:

```
make -f makefile_linux mfcompress.a
```

When building the mfcompress library, one will get the following message: "warning decimal constant is so large that it is unsigned". The C compiler is objecting to the legitimate integer constant -2147483648. This warning can be ignored since the constant is handled properly by the compiler.

5. MFLIB LIBRARY

MFLIB is a library of functions for:

a. Terminal and file I/O of parameters.
b. File I/O for DREA standard binary files by using the DBIO and MTAPEIO libraries.
c. Processing data from DREA standard binary files.

The VMS and IRIX versions of MFLIB are described in reference [15].
The Windows NT version of MFLIB is described in reference [2].

On the Linux computer thorne, the source files for MFLIB are in the directory /home/chang/MFLIB. The MFLIB library mflib.a is created by using the file makefile_linux in this directory thus:

```
make -f makefile_linux mflib.a
```

P516762.PDF [Page: 44 of 70]

6. WIDGETLIB LIBRARY

WIDGETLIB is a library of functions which allow the user to create and interact with curses widgets (window gadgets). WIDGETLIB is used by program TPULSE (reference [16]).

On the Linux computer thorne, the source files for WIDGETLIB are in the directory /home/chang/CURSES. The WIDGETLIB library widgetlib.a is created by using the file makefile in this directory thus:

```
make linux
```

7. CONLIB LIBRARY

CONLIB is a Fortran-callable library of functions for:

a. Terminal and file I/O of parameters.
b. File I/O for DREA standard binary files.
c. Processing data from DREA standard binary files.

The platform-independent version of CONLIB which runs under VMS, SGI and Windows NT is described in reference [2].

On the Linux computer thorne, the source files for CONLIB are in the directory /home/chang/CONVOLVE. The CONLIB library conlib.a is created by using the file makefile_linux in this directory thus:

```
make -f makefile_linux conlib.a
```

8. FPSMATH LIBRARY

FPSMath is a library of routines by Floating Point Systems Inc. for performing various scalar, vector and matrix mathematical operations. The source files for the single precision (32-bit floating-point) version of this library are:

a. src32.for - Fortran routines for all systems.
b. srcunix32.for - Fortran routines for UNIX systems.
c. srcvax32.for - Fortran routines for VMS systems.
d. srcc32.c - C routines.

Porting FPSMath to Linux

Either srcunix32.for or srcvax32.for could be used with GNU Fortran (g77), but srcunix32.for was used. Some changes were made to srcc32.c for use with GNU C (gcc).

To obtain separate object modules for the FPSMath routines, it was necessary to divide the files scr32.for and srcunix32.for into smaller files, each containing a single routine. These smaller files were then compiled, and the resultant object files were placed in the FPSMath

library file fpsmath32.lib. To facilitate this process, program splitfor.c (reference [2]) was used to split the large Fortran files. Another program, getobj.c (reference [2]), was used to produce a list of object files, suitable for inclusion in a make file.

The thorne library /home/chang/FPSMATH/FPS32/libfpsmath32.a is created by using the file makefile in this directory thus:

```
make linux
```

Note that the above library is the original FPSMath library, not the variant produced by Joseph Farrell of DREA. Farrell's library has "FPS_" pre-appended to all subprogram names, and has changed all dollar signs ($) to X's in these names.

9. SASIO LIBRARY

The SASIO (Surveillance Acoustics Section I/O) library provides a command language interface for Fortran programs. The original version of SASIO was written by Edwin Chaulk, using routines written by Ann Dent and David Hally at DREA. The version of SASIO ported to Linux is a modified version produced by David Chang (references [10] and [1]).

On the Linux computer thorne, the source files for SASIO are in the directory /home/chang/SASIO. The SASIO library libsasio.a is created by using the file makefile in this directory thus:

```
make linux
```

X. PORT OF VARIOUS PROGRAMS TO LINUX

Note that this section covers only the programs that are present in the Chang directories on the VMS, SGI or NT platforms. It does not cover program PRODAT, which is normally maintained by other people. The port of PRODAT to Linux is described below in a separate section.

WARNING: If the tape drive is not turned before Linux is booted, Linux will not recognize the tape device. Bear this in mind if you wish to use any Linux programs which perform tape I/O.

1. TESTING THE PROGRAMS

These programs already existed on other platforms (VMS and IRIX). Testing consisted of ensuring that the Linux programs' output was reasonable and agreed with that generated by the VMS or IRIX programs. Many of the programs read and/or write DREA standard binary data files. Numerous test binary input files were available in the VMS and IRIX directories for user Chang. To compare binary output files, their contents were listed by using program DUMPDAT, and the resultant ASCII listing files were compared.

2. DUMPDAT

Program DUMPDAT lists the contents of standard DREA binary files. The input files can be on either disk or tape (reference [6]).

On the Linux computer thorne, the source code for DUMPDAT is in the directory /home/chang/FUTIL. The program is built by using the file makefile_linux in this directory thus:

```
make -f makefile_linux dumpdat
```

To run DUMPDAT, enter:

```
/home/chang/FUTIL/dumpdat
```

3. DIRTAPE

Program DIRTAPE produces a directory listing of files on a tape. For documentation on DIRTAPE, see references [17], [15], [2], and [1].

On the Linux computer thorne, the source code for DIRTAPE is in the directory /home/chang/FUTIL. The program is built by using the file makefile_linux in this directory thus:

```
make -f makefile_linux dirtape
```

To run DIRTAPE, enter:

```
/home/chang/FUTIL/dirtape [-f] tape_device list_file
```

where the optional "-f" flag indicates that a full or verbose listing is desired. This flag can alternatively be "-F".

Example:

```
~chang/FUTIL/dirtape /dev/nst0 mytape.lis
```

4. DIRDISK

Program DIRDISK produces a directory of user-specified disk files. For documentation on DIRDISK, see reference [1].

On the Linux computer thorne, the source code for DIRDISK is in the directory /home/chang/FUTIL. The program is built by using the file makefile_linux in this directory thus:

```
make -f makefile_linux dirdisk
```

To run DIRDISK, enter:

```
/home/chang/FUTIL/dirdisk [-fs] "disk_file_specification" list_file
```

It is important to enclose the disk file specification in quotes to prevent any wildcard characters from being expanded by the shell. The optional "-f" flag indicates that a full or verbose listing is desired. The "-s" flag indicates that subdirectories are to be included in the disk file specification. The flags can be in upper or lower case (e.g., "-FS", "-s", "-F").

If you wish to list all files in a directory, set the disk file specification to "directory_name/*", rather than "directory_name".

Examples:

```
/home/chang/FUTIL/dirdisk "*.DAT" mydat.lis
~chang/FUTIL/dirdisk -fs "cruise98/*.PWR" mypwr.lis
```

5. COPYDAT

Program COPYDAT copies data from standard DREA binary files. For documentation on the VMS and IRIX versions of COPYDAT, see references [17] and [15]. For the Windows NT version of COPYDAT, see reference [2].

On the Linux computer thorne, the source code for COPYDAT is in the directory

/home/chang/FUTIL. The program is built by using the file makefile_linux in this directory thus:

```
make -f makefile_linux copydat
```

To run COPYDAT, enter:

```
/home/chang/FUTIL/copydat
```

6. TPULSE

Program TPULSE generates test signals consisting of simulated acoustic time series data. The data can be written to standard DREA binary files on tape or disk (reference [16]).

On the Linux computer thorne, the source code for TPULSE is in the directory /home/chang/SONAR. The program is built by using the file makefile_linux in this directory thus:

```
make -f makefile_linux tpulse
```

To link one's own functions to TPULSE, one should do the following in a directory other than /home/chang/SONAR:

1. cp /home/chang/SONAR/makefile_tpuser_linux makefile
2. cp /home/chang/SONAR/tpulse_user.c tpulse_user.c
3. Modify tpulse_user.c as desired.
4. make tpulse

To run the standard version of TPULSE, enter:

```
/home/chang/SONAR/tpulse
```

N.B.: Since TPULSE uses the curses library, one should ensure that the environment variable TERM is defined before running TPULSE. For example under bash, one might enter:

```
export TERM=vt100
```

Also, TPULSE uses the filter coefficients files decim08.dat and decim32.dat. If these files are not in the current working directory, one should set the environment variable FILTERS_DIR to the directory they are in. For instance under bash:

```
export FILTERS_DIR=/home/chang/SONAR
```

7. DISPX

Program DISPX displays time series data from standard DREA binary files. For

documentation on DISPX, see references [10], [2] and [1].

On the Linux computer thorne, the source code for DISPX is in the directory /home/chang/DISP. The program is built by using the file makefile_linux in this directory thus (the following example is for the bash shell):

```
export PGPLOT_DIR=/home/chang/PGPLOT/pgplot/
make -f makefile_linux dispx
```

Since DISPX uses the PGPLOT library, the environment variable PGPLOT_DIR must be defined before running DISPX. One may also wish to set the environment variable PGPLOT_XW_WIDTH to change the size of the X window display. (See the SAPLOTX section above for more information regarding these variables.) To run DISPX, enter:

```
/home/chang/DISP/dispx
```

To link your own routines to DISPX, issue the following commands from the directory where you want to store your own version of DISPX:

1. cp /home/chang/DISP/makefile_user_linux makefile
2. cp /home/chang/DISP/usersb.f usersb.f
3. Modify usersb.f as desired.
4. Define PGPLOT_DIR. Under bash, this can be done via:
   EXPORT PGPLOT_DIR=/home/chang/PGPLOT/pgplot/
5. make dispx

8. ASCDAT

Program ASCDAT copies time series data from ASCII files to standard DREA binary files. For documentation on the VMS version of ASCDAT, see reference [3]. For the Windows NT version of COPYDAT, see reference [2].

On the Linux computer thorne, the source code for ASCDAT is in the directory /home/chang/CONVOLVE. The program is built by using the file makefile_linux in this directory thus:

```
make -f makefile_linux ascdat
```

To run ASCDAT, enter:

```
/home/chang/CONVOLVE/ascdat
```

9. GENCOM

Program GENCOM generates a Fortran SASIO command table subroutine which is used by such programs as PLTPWR (references [10] and [1]). On the Linux computer thorne, the

P516762.PDF [Page: 50 of 70]

source code for GENCOM is in the directory /home/chang/SASIO. The program is built by using the file makefile_gencom_linux in this directory thus:

```
make -f makefile_gencom_linux all
```

Before running GENCOM, the environment variable TERM must be defined. For instance, under the bash shell one might enter:

```
export TERM=vt100
```

To run GENCOM, enter:

```
/home/chang/SASIO/gencom
```

10. PLTPWR

PLTPWR is a program for which performs greyscaling and plots spectral data (references [9], [10] and [1]). An earlier version of PLTPWR had been ported to Linux in May 2000 by Serge Perry of DREA. For this contract, we ported the latest version of PLTPWR (reference [1]), after examining Perry's earlier port.

Compiling PLTPWR

Program PLTPWR requires the creation of the SASIO command table subroutine pltpwr_linux_tab.f. This can be done by entering the following bash shell commands while in the /home/chang/PLTPWR directory on thorne:

```
export TERM=vt100 (Needed only if TERM isn't defined)
../SASIO/gencom
INFIL pltpwr_linux.tab
TERM OFF
OFIL pltpwr_linux_tab.f
RUN
QUIT
```

Some editing must be applied to the resultant pltpwr_linux_tab.f file:

The statement

```
HLPFIL='/home/chang/PLTPWR/pltpwr.help'
```

should be replaced by

```
CALL SASENV('PLTPWR_HELP',HLPFIL,LENHLP)
IF (LENHLP.LE.0) HLPFIL='/home/chang/PLTPWR/pltpwr.help'
```

PLTPWR can be compiled and linked by entering the following commands from within the chang/PLTPWR directory:

bash shell:

```
export PGPLOT_DIR=/home/chang/PGPLOT/pgplot/
make linux
```

csh or tcsh shell:

```
setenv PGPLOT_DIR /home/chang/PGPLOT/pgplot/
make linux
```

Warning: The makefile in the chang/PLTPWR directory copies the script file lpr_linux to lpr_post (see next section re lpr_post). If lpr_linuxisn't a suitable print script for your particular set-up, you should modify either the makefile or lpr_linux appropriately.

PLTPWR Printing

A difficulty with automatically printing PLTPWR PostScript files is that the print command can vary among UNIX systems. Therefore the SGI IRIX and Linux versions of PLTPWR have been modified so that printing of PostScript files is done via a user-written shell script file named lpr_post. lpr_post must be within the user's PATH. Thus if lpr_post was in the current working directory, one could add the working directory to PATH via the following bash command:

```
export PATH=$PATH:./
```

The lpr_post script is invoked as follows:

```
lpr_post [-P printer] [-M media] filenames
```

where:

printer	The printer name or printer queue. If equal to "local", the default printer is used.
media	Describes the media to use ("legal", "letter" or "ledger").
filenames	The names of the PostScript files to print.

Example:

```
lpr_post -P rm362a -M letter TEST.PS
```

Some sample scripts for SGI IRIX (lpr_sgi) and Linux (lpr_linux) have been supplied. These can be modified to suit one's particular set-up, and the resultant file can be copied to lpr_post.

Running PLTPWR

Before running PLTPWR, one must define several environment variables. The following examples show how to define these variables under bash:

1. TERM must be defined, if this has not already been done (e.g., export TERM=vt100).
2. One must define PGPLOT_DIR, the directory where the PGPLOT files grfont.dat and rgb.txt are found (e.g., export PGPLOT_DIR=/home/chang/PGPLOT/pgplot/).
3. One may optionally wish to define the PGPLOT X window width (e.g., export PGPLOT_XW_WIDTH=number where "number" is a fraction <= 1.0).
4. One may wish to define PLTPWR_HELP, the file specification for the PLTPWR help file (e.g., export PLTPWR_HELP=/home/chang/PLTPWR/pltpwr.help). Note that the file specification cannot exceed 40 characters; this is due to a limitation in DREA's SASIO package.

PLTPWR is run by entering the following command on the thorne computer:

/home/chang/PLTPWR/pltpwr

11. HETFIR

HETFIR is a program for heterodyning, filtering and/or decimating time series data from DREA standard binary files (references [5] and [2]). The source code for HETFIR is in the directory /home/chang/HETFIR on the Linux computer thorne. To compile HETFIR, enter the following command from within the chang/HETFIR directory:

make linux

To run Linux on thorne, use the following command:

/home/chang/HETFIR/hetfir

12. FIR6, OPTE AND FIRCOF

FIR6, OPTE and FIRCOF are filter design programs (reference [2]). Program FIR6 designs finite input response (FIR) digital filters by using a window function. It can produce lowpass, highpass or bandpass filters. Program OPTE designs FIR lowpass filters by using the Remez exchange algorithm. Program FIRCOF reads a FIR filter coefficients file as output by programs FIR6 and OPTE. It creates another filter coefficients file for input to program HETFIR. FIRCOF plots the filter response data, allowing for user-specified decimation, heterodyning and filter quantization.

The source files for these programs are in the directory /home/chang/FIR on the Linux thorne computer. To compile these programs, enter the following bash commmands in the chang/FIR directory:

export PGPLOT_DIR=/home/chang/PGPLOT/pgplot/
make linux

PGPLOT_DIR must defined before running either OPTE or FIRCOF (see above). The three programs can then be run via the following commands:

```
/home/chang/FIR/fir6
/home/chang/FIR/opte
/home/chang/FIR/fircof
```

XI. PORT OF PRODAT TO LINUX

1. BACKGROUND

PRODAT is a program for performing spectral analysis of time series data. The source code for PRODAT is stored in a repository which is governed by the CVS version control system (reference [PRODAT 99]). Two major releases of PRODAT that are present in this repository are:

1. RELEASE_322151 (January 19, 1999): This is the "main trunk" version of PRODAT currently used on operating systems other than Windows NT.
2. RELEASE_2000_MAR_28 (March 23, 2000): This is a "branch" version of PRODAT used under Windows NT. It is not based on the above-mentioned "main trunk" release, but is instead derived from the November 26, 1998 version of the main trunk.

In addition to these two "official" releases, Serge Perry of DREA modified RELEASE_322151 in May 2000 to produce a version of PRODAT that would run under Linux on Intel X86 machines. Perry's version lacked several features (e.g., tape input and printer control), and was not stored in the CVS repository.

For the Linux port of PRODAT under this contract, we created another CVS branch of PRODAT (LINUX_X86). This branch is based on the main trunk (RELEASE_322151), but incorporates features from the NT branch and from Perry's version. Chris Maxwell (the owner of the PRODAT repository) did not allow other users to modify the PRODAT CVS repository on the SGI computer warrior. Therefore we copied Maxwell's CVS repository to the thorne computer, and worked with the thorne copy.

2. CVS REPOSITORY

The Linux PRODAT CVS repository is stored in the directory /home/chang2/CVSROOT on the thorne computer. Currently only user "chang2" can access the repository on thorne via CVS, but anyone can make their own copy of the repository. (The reason for this limitation was to prevent other users from damaging the chang2 repository. CVS requires write access to the repository, even if you're only trying to read its source files.) To access the repository via CVS, one must first define the repository's location:

```
export CVSROOT=/home/chang2/CVSROOT
```

Obtaining and compiling the software in the repository can be done as follows:

PRODAT

Assume that the environment variable MY_PRODAT defines the directory where you want

to store the PRODAT code.

```
cd $MY_PRODAT
cvs export -r LINUX_X86 src/Prodat
cd src/Prodat
(Now read the file read.me and ensure that the file machine.typ
 has the appropriate contents.)
make
chmod ugo+x prodat.sh
```

To set up the PRODAT Front End (PFE):

```
cd $MY_PRODAT/src/Prodat/pfe
cp linux.mak platform.mak
make all
```

The PFE uses the file run_prodat in the user's PATH to run PRODAT. One way to define run_prodat:

```
ln -s $MY_PRODAT/src/Prodat/prodat/prodat.sh run_prodat
```

PRODAT Test Suite (PTS)

Assume that the environment variable MY_PTS defines the directory where you want to store the PTS code.

```
cd $MY_PTS
cvs export -D 2001-03-01 src/TestSuite/Prodat
cd src/TestSuite/Prodat
mkdir output
cd tools
cp linux.mak platform.mak
make all
```

Note that the PTS CVS repository does not contain the binary data files used by the test suite. The missing files are:

1. base*.* files for the $MY_PTS/src/TestSuite/Prodat/baseline directory.
2. *.dat files for the $MY_PTS/src/TestSuite/Prodat/input directory.

The missing files can be obtained from the corresponding subdirectories of /home/chang2/Prodat_Testing/src/TestSuite/Prodat on thorne, or from the subdirectories of ~murray/TestSuite/Prodat on the SGI machine warrior.

3. PORTING ISSUES

Some problems encountered during the Linux port were:

1. PRODAT converted a power value of 0.0 volts**2 to 0.0 dB (subroutine FPS_VDBPWR in the FPSMath library). This was changed so that 0.0 volts**2 would be converted to -395.691 dB, as per other code in PRODAT.

2. The functions used to convert floating-point numbers between various formats did not work correctly under Linux. These functions assumed that a floating-point assignment of the form A=B would always result in A being an exact bitwise copy of B. However, this assumption was not true for the GNU C (gcc) and GNU Fortran (g77) compilers. To correct this problem, the relevant floating-point assignment statements were replaced with corresponding integer assignments.

3. It was found that the Windows NT version of PRODAT wrote all output files in big endian IEEE format. However, NT PRODAT could only process input spectral data files if they were in native format (little endian IEEE). To avoid having Linux PRODAT produce files it cannot read, Linux PRODAT outputs spectral data in native format.

4. It was noted that PRODAT cannot handle more than one INFILE command for tape input. Apparently once PRODAT opens a tape, it does not close it. Consequently, when a second INFILE command is issued for the tape, PRODAT finds that the tape is already open and issues an error message. This is a design feature of PRODAT rather than a porting issue, so nothing was done about this.

4. TESTING PRODAT

To check the operation of Linux PRODAT, use was made of Prior Data Sciences' PRODAT Test Suite (PTS). The files for this test suite can be found in the thorne directory:

/home/chang2/Prodat_Testing/src/TestSuite/Prodat

For further information on the PTS, see the file README in the above directory.

There are 13 tests in the PTS, but tests 12 (A-D data acquisition) and test 13 (network input) were not done since Linux PRODAT does not handle these types of input. Test 10 (clutter tool) was not performed because no information was given on what this test entailed, and we were unfamiliar with the clutter tool. Following is the outcome of the other tests:

1. Test 1: No errors.

2. Test 2: The message "WARNING: Maximum selected frequency too high!"

appeared several times. It is assumed that this is a normal occurrence.

3. Tests 3 and 4: No errors. The x-axis tick mark labels on the time series plots overlapped each other. Since the labels were in units of absolute time and were one second apart, this problem was not unusual for a long x-axis.

4. Test 5: There was one error condition and 7 warning conditions out of 734208 values, when comparing the Linux PRODAT output .BRG file with the baseline data file. (The PTS program signals an error condition for samples which differ by >= 10 percent; otherwise it issues a warning for samples which differ by >= 0.25 percent.) None of the errors and warnings for test 5 were judged to be significant since they signaled differences between small numbers.

5. Test 6: There were two error conditions and 22 warning conditions out of 1468416 samples when comparing the output .CRS file with the baseline file. None of these conditions were judged to be significant; they were differences in small phase angles.

6. Test 7: This was similar to test 6, except that .COH files were compared. Error and warning conditions were identical to those reported for test 6. The check list for test 7 differs slightly from that of test 6. It is believed that this is a misprint, since tests 6 and 7 display the same number of plots.

7. Test 8: No errors.

8. Test 9: The PTS program indicated that there was a "problem with hdr.pwr.centerfreq". The center frequency of the first bin for the output .PWR file. Since single precision floating-point calculations are accurate only to 7 significant figures, the "problem" was insignificant. Although the check list for test 9 indicates that there are supposed to be five grams displayed, six grams appeared on the screen. It is assumed that the check list is in error.

9. Test 11: To create an input tape for this test, the input test file from test 1 (puretone.dat) was copied to tape using program COPYDAT. Three copies in little endian IEEE, big endian IEEE, and VAX number format were written to the tape. The test 1 baseline output file base01.pwr was copied to base11.pwr. Linux PRODAT was tested with each of the three tape files and no errors were detected.

5. RUNNING PRODAT

Configuring X

The SDU Server program used by PRODAT requires that the video display be set for a colour depth of 8 (i.e., 256 colours). This can be done for the local display by modifying the file

/etc/X11/XF86Config. In the "Screen" sections at the end of this file add the line

DefaultColorDepth 8

after ensuring that there is a "Display" subsection for that colour depth. (The DefaultColorDepth keyword is used in XFree86 3.3.x; for XFree86 4.0 use the DefaultDepth keyword instead.) For further details see "man XF86Config" or the web page

http://home.mdt.net.au/~john/cds/htm/config-x.htm
(This web page is for "Huiac" Debian Linux, but the discussion of XF86Config also applies to Red Hat Linux.)

Note that you must reboot Linux for the changes in XF86Config to take effect.

SDU Server needs to allocate some colours under the X Window system. However, in 256 colour mode, the Gnome desktop allocates all available colours, so SDU Server cannot run. Moreover some programs (e.g., Netscape) may use up a lot of colors. To resolve these problems, use another window manager like KDE or AfterStep, and be cautious regarding which programs you run simultaneously with SDU Server.

PRODAT Printing

Automatically printing PRODAT PostScript files is done via the shell script lpr_post, which must be within the user's PATH. This script is call compatible with the PLTPWR lpr_post script (see section X.10. above).

Text Interface to PRODAT

To run PRODAT using a text mode interface, enter the command

/home/chang2/Linux_Prodat/src/Prodat/prodat.sh

(Note that the DISPLAY environment variable must be defined before running PRODAT, otherwise PRODAT will crash.)

PRODAT GUI

The PRODAT Front End (PFE) provides a PRODAT GUI. Output from the PFE is piped to the run_prodat command, so one should set up an appropriate run_prodat link file within one's current PATH. For instance:

ln -s /home/chang2/Linux_Prodat/src/Prodat/prodat.sh run_prodat

Once run_prodat has been defined, one can start the PFE via:

/home/chang2/Linux_Prodat/src/Prodat/pfe/pfe

When the PFE is run, a menu bar will appear as well as the usual PRODAT text window. Note that cannot enter commands directly into the PRODAT text window; all input is via the menu bar. Some PRODAT features are not available via the PFE and vice versa. For instance, data input can only be from disk (not tape) files. Although "File/Input Source/Network" appears in the PFE menu, Linux PRODAT does not support network input.

The PFE can also be used to generate PRODAT macro files. For instance, the following two commands will pipe PFE's output to the file macrofile.mac:

ln -s /bin/cat run_prodat
/home/chang2/Linux_Prodat/src/Prodat/pfe/pfe > macrofile.mac

P516762 PDF [Page: 60 of 70]

XII. APPENDIX A - INSTALLING LINUX ON A WINDOWS 2000 DESKTOP SYSTEM

Linux was installed on a Dell Precision WorkStation 420 MT desktop computer (swifty.drea.dnd.ca) with the following hardware:

1. Pentium III processor, 933 MHz.
2. 256 MB ECC RDRAM memory.
3. 256 KB level 2 cache.
4. NVIDIA GeForce2 GTS 32 MB DDR video controller.
5. Sound Blaster Pro compatible card (Crystal Sound Fusion).
6. 3Com 3C920 integrated fast ethernet controller (3C905C-TX compatible).
7. Integrated Ultra 160 SCSI controller.
8. Integrated ATA/66 EIDE controller.
9. Two 40 GB EIDE hard disk drives. Primary drive 0 has model number WDC WDB400BB-75AUA1, whereas primary drive 1 has model number WDC WDB400BB-00AUA1.
10. IDE/ATAPI CD-RW (Sony CRX140E, 8x/4x/32x - secondary drive 0).
11. IDE/ATAPI Iomega Zip 250 MB removable media drive (secondary drive 1).
12. 1.44 MB floppy disk drive.
13. PS/2 Logitech compatible three button mouse.
14. 104 key keyboard.
15. Dell P991 19" Flat Display Trinitron monitor.

The operating system was Microsoft Windows 2000 Professional.

1. ATTEMPTING TO INSTALL LINUX ON THE WINDOWS DISK

Consideration was given to installing Linux on the same disk as Windows 2000. However, the Linux boot partition must be in the first 1024 cylinders of the disk (approximately the first 8 GB since the cylinder size is 8,225,280 bytes for this particular disk; see reference [4].) Therefore it would be necessary to shrink the Windows 2000 partition to about 7.5 GB. We attempted to make room for Linux by using the standard Windows 2000 disk defragmentation tool (a version of Diskeeper 5), but it left a system file in the middle of the disk. We next downloaded a trial version of Diskeeper 6 and used it to do a boot time defragmentation of the disk. This also did not move the offending system file, which we believe is part of the Master File Table (MFT). It should be noted that moving some system files (e.g., the MFT and paging file) is best done at boot time, since doing it while Windows is running may not be safe. As it did not appear that we could shrink the Windows 2000 partition sufficiently, we decided to install Linux on a separate disk.

2. ATTEMPTING TO INSTALL RED HAT LINUX 6.0

For compatibility with the thorne installation, we decided to try to install Red Hat Linux 6.0 rather than Linux 7.0. However, the installation was not successful:

1. The ethernet card was not recognized.
2. The video card was not recognized.
3. The disk partition created to hold the /home directory had problems - inode had illegal blocks. When we looked at the disk using the Windows 2000 Disk Management tool, we found that the /home partition was smaller than expected; there was about 1.38 GB free space at the end of the disk.

3. INSTALLING RED HAT LINUX 7.0

The installation followed the method used to put Red Hat Linux 6.0 on thorne.drea.dnd.ca (see section II. above). Rather than repeat that information here, we will only mention features peculiar to the swifty.drea.dnd.ca installation. Note that neither Motif nor IDL was installed on swifty. Also a printer was not configured for swifty.

The Red Hat Linux 7.0 default installer differs from the Linux 6.0 installation program; the former is GUI based, whereas the latter has a text interface. However both installers follow basically the same steps.

3.1 Preparing for Installation

As it turned out, the Linux installation program was able to automatically identify most of the hardware. The information that we had to provide was:

1. The mouse's type and number of buttons (generic PS/2 with three buttons). (The mouse was supposedly Logitech compatible, but we did not know which Logitech model it was compatible with.)

2. Whether or not the disk drives were accessed in LBA (linear block addressing) mode, sometimes referred to as "large disk" mode. We were unable to determine if the disk drives used LBA mode. It was presumed that they did use LBA mode since they were new, large drives.

3. Network-related information:

   IP address
   Netmask
   Network
   Broadcast
   Hostname: swifty.drea.dnd.ca
   Gateway IP address (router)
   Name server IP addresses

Note: "Network" and "broadcast" are peculiar to the Linux 7.0 installation, and are not used in a Linux 6.0 installation.

3.2 Partitioning the Disks

Prior to the Linux installation, the disk drives were partitioned as follows. The first EIDE drive was divided into two primary partitions: a 15 MB FAT partition (MB=1,048,576 bytes), and a 38146 MB NTFS partition. According to the Windows 2000 Disk Management tool, the FAT partition is for "EISA Configuration". The Windows 2000 Professional operating system was located in the NTFS partition. The second EIDE drive was a new one and contained nothing.

We decided to make no changes to the first EIDE hard drive, and to partition the second hard drive as follows:

Table 8. *Partitioning of the second EIDE Drive on Windows 2000 machine*

Type:	FAT root (/)	Linux swap	Linux swap	Linux swap	Linux /usr	Linux /home	Linux
Size (MB):	500	532	125	125	125	1500	Rest of disk.

The first two partitions are primary ones. The swap, /usr and /home partitions are logical partitions within an extended partition. The / and /usr partitions are larger than those used for the thorne installation, since Red Hat Linux 7.0 requires more disk space.

Prior to repartitioning the second disk, a Windows 2000 emergency repair disk was created. The Windows 2000 Disk Management tool was used to create the FAT partition on the second drive. Creation of the Linux partitions was done later via the Disk Druid program (see below).

3.3 Installing Linux

The installation process was generally straightforward. Special steps for our particular installation are mentioned in the following paragraphs.

Creating Partitions for Linux

Partitioning via Disk Druid resulted in the following set-up:

Table 9. Creating disk partitions for Linux.

Mount Point	Device	Actual Size	Type
<not set>	hda1	15M	0xde
<not set>	hda2	38146M	NTFS/HPFS
/fat1	hdb1	502M	DOS 16-bit >= 32M
/	hdb2	533M	Linux native
/usr	hdb5	1506M	Linux native
<swap>	hdb6	125M	Linux swap
<swap>	hdb7	125M	Linux swap
<swap>	hdb8	125M	Linux swap
/home	hdb9	35244M	Linux native

Disk Druid put the /usr partition before the swap partitions, even though we specified the /usr partition after the swap partitions. We intentionally did not mount any partitions containing the Windows 2000 system files, to reduce the possibility that Linux users might damage these files. Any file sharing between Windows 2000 and Linux can be done via the /fat1 partition on the second hard drive.

Selecting Components

Red Hat Linux 7.0 has a somewhat different component list than Linux 6.0. In addition to the default components, we included the following:

1. KDE
2. DOS/Windows Connectivity
3. Graphics Manipulation
4. Authoring/Publishing
5. EMACS
6. Development
7. Utilities

KDE was included due to the problems with running PRODAT under GNOME.

X Configuration

Under "Monitor Configuration", there wasn't an entry for our particular monitor (Dell Ultrascan P991), but under "DDC Probed Monitor - DCL 5178", it did report the proper horizontal sync rate (30 -107 kHz) and vertical sync rate (48 - 120 Hz). For our video card, it selected "NVIDIA GeForce 2 (generic)" with 32768K memory. There was an entry for a "NVDIA GeForce DDR (generic)", but since the GeForce 2 setting seemed to work all right, we didn't try the DDR setting.

Preparing the Computer for Dual Booting

This follows the same method as for dual booting Windows NT and Linux. See reference [4].

When the Linux LILO boot loader is invoked, it presents two options: linux (default) and linux-up. The former is for a symmetric multi-processor (SMP) version of Linux, whereas the latter is for a uni-processor version. The Dell WorkStation 420 is an SMP machine, although only one processor is installed in swifty.

Linux Configuration

a. In order to use the SCSI interface on this machine, the SCSI controller had to be turned on. This was done by changing the integrated controller settings in the BIOS. (Pressing F2 at system start up allows one to access the BIOS menus.) Once we had turned on the controller, Linux 7.0 recognized at boot time that we had added new hardware. It successfully identified the SCSI controller and loaded the appropriate driver.

b. The default Linux 7.0 installation caused Linux to start up in text mode. To start Linux in X Window mode, we changed the following line in the /etc/inittab file from

```
id:3:initdefault:
```

to

```
id:5:initdefault:
```

c. The "top" command seemed to indicate that the Linux kernel had access to all of memory. Therefore, unlike in the saturn machine case, we did not add the following line to the "per image" section of the /etc/lilo.conf file:

```
append="mem=256M"
```

d. An ordinary user was able to mount the floppy disk or CD ROM, so no additional work was required to make this possible.

e. Unlike in the Linux 6.0 case, the message "Can't locate module st" did not occur during the boot process, so no corrective action was necessary.

f. To permit an ordinary user to access a SCSI tape drive (if it was installed), we changed the permissions for some files in the /dev directory:

```
chmod o+rw /dev/st0
chmod o+rw /dev/nst0
```

g. Fortran had been automatically installed by the Linux 7.0 installer, so we did not have to install it ourselves.

h. To permit remote telnet and FTP access to the computer, we installed the following packages via GnoRPM: telnet-server, wu-ftpd, and xinetd (Linux 7.0 Binary CD 1).

i. To allow SCSI tape support, we installed the mt-st package (Linux 7.0 binary CD 1) by using GnoRPM. Unlike in the Linux 6.0 case, this package had not been automatically installed; this may be because we did not have the SCSI controller turned on when we did the Linux 7.0 install. As in the Linux 6.0 case, one must turn on the tape drive before booting Linux in order to use the drive.

j. The "whatis" database had already been created by the Linux 7.0 installer. However, we had to update the database because we had subsequently installed more packages. The update was done via:

/usr/sbin/makewhatis -u

k. The sound card had already been configured during the install, but the card did not appear to work under Linux. (It worked under Windows 2000 though.) We ran sndconfig in an attempt to correct the problem. Sndconfig identified the sound card, but when it played back a sample sound file we did not hear anything.

P516762.PDF [Page: 66 of 70]

References

As noted below, several of the following reports are currently available as ASCII files on the DREA FIRM VMS cluster.

1. Chang, David, "Final Report and User's Manual for Changes and Updates to Acoustic Analysis Software", DREA Contractor Report (Draft), October 2000 (available in the file FIRM::PS_72:[CHANG.FIR]AAS_UPDATE.PS).

2. Chang, David, "Final Report and User's Manual for Changes and Updates to Acoustic Data Manipulation Software", DREA Contractor Report (Draft), September 1999 (available in the file FIRM::PS_72:[CHANG.FIR]ADMS_UPDATE.PS).

3. Chang, David, "User's Manual for Data Convolution Software", DREA Contractor Report (Draft), April 1996 (updated September 1999) (available in the file FIRM::PS_72:[CHANG.CONVOLVE]CON_DRAFT.TXT).

4. Blackshaw, Bruce P., "Dual Booting Linux and Windows 2000 on Large Hard Disks", http://www.enterprisedt.com/publications/dual_boot.html, January 27, 2001.

5. Chang, David, "User's Manual for Data Filtering and Heterodyning Software", DREA Contractor Report (Draft), December 1994 (updated September 1999) (available in the file FIRM::PS_72:[CHANG.HETFIR]HETFIR_DRAFT.TXT).

6. Chang, David, "User's Manual for Programs to Interpolate between Spectral Data, to List Data Files, and to Merge Impact Data Files", DREA Contractor Report (Draft), March 2000 (updated January 2001) (available in the file FIRM::PS_72:[CHANG.IMPACT]IMPACT_ETC.PS).

7. Chang, David, "Final Report on the Installation of Linux and Other Software on a Windows NT System", DREA Contractor Report (Draft), October 1999 (available in the file FIRM::PS_72:[CHANG.UTIL]LINUX.PS).

8. Chang, David, "User's Manual for Active Sonar Acoustic Data Matched Filtering and Clustering Software", DREA Contractor Report (Draft), December 1995 (available in the file FIRM::PS_72:[CHANG.SONAR]MF_USER_DRAFT.TXT).

9. Farrell, Joseph B., "PLTPWR: A Versatile Tool for Greyscale and Power Spectral Display", DREA Technical Communication 92/306, April 1992.

P516762.PDF [Page: 67 of 70]

10. Chang, David. "Final Report and User's Manual for Modifications to PLTPWR, DISP, DISPX and PROPWR", DREA Contractor Report (Draft), March 1995 (updated September 1999) (available in the file FIRM::PS_72:[CHANG.PLTPWR]PLTPWR_ETC_USER.PS).

11. Maxwell, C. "PRODAT Enhancement and Repair", DREA Contractor Report CR 1999/048, May 1999.

12. Farrell, Joseph B., "PROPWR: A Versatile Tool for 1/3 Octave Processing and Narrowband Line Tracking", DREA Technical Communication 93/303, February 1993.

13. Red Hat Software Inc., "The Official Red Hat Linux 6.0 Installation Guide", Durham, North Carolina, 1999.

14. Staal, Philip R., "SAPLOT Scientific Graphic Software History and Overview", DREA Technical Communication, 1989.

15. Chang, David, "Technical Manual for File Utility Software", DREA Contractor Report (Draft), August 1999 (available in the file FIRM::PS_72:[CHANG.FUTIL]FUTIL_TECH_DRAFT.PS).

16. Chang, David, "Program TPULSE for Generating Simulated Acoustic Data", DREA Contractor Report, April 1998 (Updated March 2001) (available in the file FIRM::PS_72:[CHANG.SONAR]TPULSE_DRAFT.PS).

17. Chang, David, "User's Manual for File Utility Software", DREA Contractor Report (Draft), March 1999 (updated January 2001) (available in the file FIRM::PS_72:[CHANG.FUTIL]FUTIL_USER_DRAFT.PS).

P516782.PDF [Page: 68 of 70]

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1 ORIGINATOR (the name and address of the organization preparing the document Organizations for whom the document was prepared, e g Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8)

Evans Computer Applications Ltd.
6424 Norwood St
Halifax, NS B3H 2L3

2 SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable)

UNCLASSIFIED

3 TITLE (the complete document title as indicated on the title page Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title)

Final Report on the Port of Signal Processing Software to Linux

4 AUTHORS (Last name, first name, middle initial If military, show rank, e g Doe, Maj John E)

David R. Chang

5 DATE OF PUBLICATION (month and year of publication of document)	6a NO OF PAGES (total containing information Include Annexes, Appendices, etc)	6b NO OF REFS (total cited in document)
August 2001	53	17

7 DESCRIPTIVE NOTES (the category of the document, e g technical report, technical note or memorandum If appropriate, enter the type of report, e g interim, progress, summary, annual or final Give the inclusive dates when a specific reporting period is covered)

CONTRACTOR REPORT

8 SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development Include address)

Defence Research Establishment Atlantic
PO Box 1012
Dartmouth, NS, Canada B2Y 3Z7

9a PROJECT OR GRANT NO (if appropriate, the applicable research and development project or grant number under which the document was written Please specify whether project or grant)	9b CONTRACT NO (if appropriate, the applicable number under which the document was written)
Project 11CC	W7707-0-8308

10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity This number must be unique to this document)	10b OTHER DOCUMENT NOs (Any other numbers which may be assigned this document either by the originator or by the sponsor)
	DREA CR 2001-108

11 DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12 DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected)

Unlimited

P516762.PDF [Page: 69 of 70]

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual)

This report describes how the Red Hat Linux 6.0 operating system was installed on a Windows NT system at DREA. The subsequent port of some signal processing software to Linux, including SAPLOTX, PRODAT, PLTPWR, PROPWR, DISPX, HETFIR, and COPYDAT, is also outlined. Additionally, the report describes the difficulties encountered while performing these tasks, and the issues that still remain to be resolved.

As an auxiliary task, Red Hat Linux 7.0 was installed on a Windows 2000 system. This activity is also fully documented.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title)

signal processing software
Red Hat Linux operating system
Windows NT
SAPLOTX software
PROPWR software
COPYDAT software
TPULSE software
DISPX software
PLTPWR software
HETFIR software
PRODAT software

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM